

**Enuntziatua****1. ARIKETA (3 puntu) Kate osagarriak (karaktere-kateak erabiliz)**

ADN molekula bakoitza bi kate osagarriaz eratuta egon ohi da eta kate horietako bakoitzak nukleotido izeneko konposatu kimiko ugari izaten ditu. Nukleotidoak, besteak beste, adenina (A), guanina (G), timina (T) eta zitosina (Z) elementuez eratuta daude (elementu horiei oinarrizko elementuak deituko diegu). Kate bakoitzeko oinarritzko elementu bakoitza azukre molekula bati elkartuta egongo da eta, hidrogenozko lotura bati esker, molekulako beste katean aurkitzen den oinarritzko elementu osagarri bati ere lotuta egongo da. Oinarritzko elementuen arteko afinitate kimikoa dela eta, adenina (A) duten nukleotidoak beti timina (T) dutenekin elkartzen dira eta zitosina (Z) dutenak guanina (G) dutenekin elkartzen dira.

Esate baterako ADN molekula bat osatzen dutenetako kate batean AGATZ sekuentzia balego, molekulako beste katean bere sekuentzia osagarria TZTAG edukiko litzateke.

Jarraian zehazten dena egin behar da:

- a) (0,75 puntu) 'A', 'G', 'T' eta 'Z' karaktereez osatutako kat1 kate bat emanda, kat2 bere kate osagarria kalkulatzeko duen funtzioa definitu, 'A'-ren osagarria 'T' dela eta 'G'-ren osagarria 'Z' dela kontuan hartuz:

**void sekuentzia\_osagarria(char kat1[ ], char kat2[ ]);**

- b) (0,5 puntu) Kate bat emanda, katean zebat 'G' eta zenbat 'Z' dauden kalkulatzeko balio duten bi funtzio definitu:

**int g\_kopurua(char kat[ ]);**

**int z\_kopurua(char kat[ ]);**

- c) (puntu 1) Adenina eta timina hidrogenozko bi zubiren bidez elkartzen dira; zitosina eta guanina, berriz, hiru zubiren bidez. Horregatik, zenbat eta Z-G edo G-Z lotura gehiago eduki, hainbat eta egonkorragoa izango da ADN molekula bat. ADN molekula baten egonkortasuna ezagutzeko ADN molekula hori osatzen duten 2 kateak ezagutzea ez da beharrezkoa, batekin nahikoa dugu, kateak osagarriak baitira.

Bi kate emanda, egonkorragoa zein den erabakitzen duen funtzioa definitu, aurreko atalean definitutako *g\_kopurua* eta *z\_kopurua* funtzioak erabiliz. Kate bakoitza ADN molekula desberdin batena izango da. Kate bat beste bat baino egonkorragoa izango da 'Z' kopurua gehi 'G' kopurua handiagoa bada. Funtzioak 1, 2 edo 0 balioa itzuli beharko du:

- lehenengo katea bigarrena baino egonkorragoa bada, 1 balioa itzuliko du.
- bigarrena lehenengo baino egonkorragoa bada, 2 balioa itzuliko du.
- Egonkortasun maila bera badute, 0 balioa itzuliko du.

**void egonkorra(char kat1[ ], char kat2[ ]);**

**(hurrengo orrian jarraitzen du →)**

**d)** (0,75 puntu) Programa nagusia idatzi. Programa nagusiak honako ekintza hauek burutu beharko ditu:

- 'A', 'G', 'T' eta 'Z' karaktereez osatutako kate bat eskatu erabiltzaileari. Kateak gehienez 100 karaktere izan ditzake. Kate horretan 'A', 'G', 'T' eta 'Z' karaktereak edozein ordenetan eta edozein kopurutan ager daitezke eta beste karaktererik ez dela egongo, hau da katea egokia izango dela suposatu behar da. Erabiltzaileak ez du ezer egiaztatu behar.
- Kate osagarria kalkulatu eta pantailan aurkeztu.
- Beste ADN molekula bati dagokion katea eskatu (hau ere egokia izango dela eta gehienez 100 karaktere izango dituela suposatuz) eta lehenengo katea kontuan hartuz egonkorrena zein den erabaki erantzuna pantailan aurkeztuz.

**Exekuzio-adibidea:** (erabiltzaileak teklatutakoa letra etzanez eta azpimarratuta ageri da)

Sartu ADN molekula bati dagokion katea: <u>AAGZTZZA</u> Kate horren osagarria honako hau da: TTZGAGGT Sartu beste ADN molekula bati dagokion katea: <u>GGZZZ</u> 2. ADN molekula egonkorragoa da. Sakatu tekla bat amaitzeko.
---

Pantaila

Adibide honetan bigarren ADN molekula lehenengoa baino egonkorragoa da bigarreneko G kopurua gehi Z kopurua (5) lehenengoko G kopurua gehi Z kopurua (4) baino handiagoa delako.

**2. ARIKETA (4 puntu) Munduko futbol txapelketa (karaktere-kateak eta bektoreak erabiliz)**

Munduko futbol txapelketari buruzko informazioa gordetzeko *Selekzioak* egitura dugu:

'T'	't'	'a'	'A'	'y'	'Y'	'S'	's'	'p'	'P'	'i'	'I'	'n'	'N'	'G'	'g'	'e'	'E'	'c'	'C'	'u'	'U'	'k'	'K'	'0'	...
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Selekzioen izenak karaktere-kate batean gordetzen dira, komaz bananduta. Gehienez 32 talde egon daitezke.

Selekzio bakoitzaren irabazitako, berdindutako eta galdutako partidu-kopuruak gordetzeko hiru taula (*irabazitakoak*, *berdindutakoak*, *galdutakoak*) erabiliko dira. Horrela, adibidea jarraituz, taula hauen 0. posizioek Italiak irabazitako, berdindutako eta, hurrenez hurren, galdutako partidu kopuruak gordeko dituzte, 1. posizioetan Espainiar selekzioaren datuak egongo dira, eta horrela azken selekziora arte (adibidean 3. posizioa, UK-ko selekzioari dagokiona).

Burutu beharrekoak honako hauek dira:

**a) (0,25 puntu)**

Selekzioen izenak dituen katea emanda, zenbat selekzio dauden kalkulatzeko duen **funtzioa** idatzi. Izenak koma batez bananduta egongo dira:

**int selekzio\_kopurua (char selek[ ]);**

**b) (0,5 puntu)**

Irabazitako, berdindutako eta galdutako partidu-kopuruei buruzko datuak dituzten taulak eta selekzio-kopurua emanda, beste taula batean selekzio bakoitzak lortutako puntuak gordetzen dituen **funtzioa** idatzi. Irabazitako partidu bakoitzeko 3 puntu lortzen dira, berdindutako partidu bakoitzeko puntu 1 eta galdutako partiduek punturik ez dute ematen:

**void puntuak\_kalkulatu (int irabazi[ ], int berdindu[ ], int galdu[ ], int puntuak[ ], int selek\_kop);**

**c) (3,25 puntu) (0,25 + 0,25 + 1 + 0,25 + 1,5)**

Funtzio nagusia idatzi. Funtzio nagusiak honako urrats hauek jarraitu beharko ditu:

- Selektzioen izenez osatutako karaktere-katea eskatu eta jaso. Izenak komaz bereiztuta egongo dira eta gehienez 250 karaktere izango dira. Ez da ezer egiaztatu beharko, dena egokia dela suposatuta behar da.
- *selekzio\_kopurua* funtzioa erabiliz zenbat selekzio dauden kalkulatu.
- Selektzio bakoitzaren irabazitako, berdindutako eta galdutako partidu-kopuruak eskatu erabiltzaileari.
- *puntuak\_kalkulatu* funtzioa erabiliz selekzio bakoitzaren puntuazioa kalkulatu.
- Puntuazioa kontuan hartuz bi selekzio onenak zein diren kalkulatu eta bi selekzio horien izenak eta puntuak aurkeztu pantailan.

**Exekuzio-adibidea:** (erabiltzaileak teklatutakoa *letra etzanez eta azpimarratuta* ageri da)

```
Sartu selekzioen izenak: Italy,Spain,Greece,UK
Sartu 1. selekzioaren irabazi, berdindu eta galdutako
partiduak: 3 0 1
Sartu 2. selekzioaren irabazi, berdindu eta galdutako
partiduak: 1 2 1
Sartu 3. selekzioaren irabazi, berdindu eta galdutako
partiduak: 2 0 2
Sartu 4. selekzioaren irabazi, berdindu eta galdutako
partiduak: 1 0 3
Taldea onenak Italy eta Greece dira 9 eta 6 punturekin.
```

Pantaila

**3. ARIKETA (3 puntu) Blum-en osoak**

Blum-en zenbaki osoak kriptografian eta konplexutasunaren teorian erabiltzen dira zenbaki hauek maneiatzen dituzten algoritmo eraginkorrak aurkitzea oso zaila delako.

Osoa eta positiboa den  $n$  zenbaki bat Blum-en osoa dela esango dugu jarraian aipatzen diren lau baldintzak betetzen dituzten  $p$  eta  $q$  bi zenbaki oso existitzen badira:

- $n = p * q$
- $p$  eta  $q$  zenbaki lehenak dira.
- $p$  eta  $4$ ren arteko zatiketaren hondarra 3 da.
- $q$  eta  $4$ ren arteko zatiketaren hondarra 3 da.

Beste era batera esanda,  $n$  Blum-en osoa izango da lehena den eta zati 4 egiterakoan hondar bezala 3 ematen duen  $n$ -ren  $p$  zatitzaile bat existitzen bada eta gainera  $n$  zati  $p$  egiterakoan lortzen den  $q$  zenbakia ere lehena bada eta  $q$  zati 4 egiterakoan hondar bezala 3 lortzen bada.

**Adibideak:**

- ✓ 133 Blum-en osoa da:  $133 = 7 * 19$  da, 7 eta 19 lehenak dira eta 7 eta 19 zati lau egiterakoan hondarra 3 da.
- ✓ 217 ( $217 = 7 * 31$ ), 301 ( $301 = 7 * 43$ ) eta 589 ( $589 = 19 * 31$ ) ere Blum-en osoak dira, 7 eta 19ren kasuan bezala 31 eta 43 lehenak baitira eta zati lau egiterakoan hondarra 3 baita.

Honako hau egin behar da:

**a) (1,5 puntu)**

**int lehena**43(**int x**) funtzioa definitu. Funtzio honek 1 itzuliko du  $x$  zenbaki lehena bada eta  $x$  zati 4 egitean hondarra 3 bada eta bestela 0 itzuliko du.

**b) (1,5 puntu)**

Programa nagusia idatzi. Bertan erabiltzaileari zenbakiak eskatuko zaizkio Blum-en osoak al diren ala ez erabakitzeko. Programak balio bat eskatu ondoren Blum-en zenbakia al den ala ez adieraziko du eta beste balio bat sartu nahi al duen galdetuko dio erabiltzaileari. Galdera honi erantzuteko 'b' edo 'e' balioak besterik ez dira onartuko. Ondoren, baiezko erantzuna jaso bada ('b') beste zenbaki bat eskatuko da, prozesu osoa erabiltzaileak ezezko erantzuna eman arte errepikatuz. Ezezko erantzuna jasotzen denean programa nagusia amaitu egingo da.

**OHARRA: Zenbaki lehenak** unitatearekin (1) eta bere buruarekin bakarrik zatigarriak diren zenbaki positibo osoak dira, baina definizioz 1 ez da lehena. Aurreneko hogeit hamar zenbaki lehenak 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67 eta 71 dira. Beste era batera esanda, zenbaki lehenak justu bi zatitzaile dituzten zenbakiak dira. Zatitzaile gehiago edo gutxiago dituztenak ez dira lehenak.

**Exekuzio-adibidea:** (erabiltzaileak teklatutakoa letra etzanez eta azpimarratuta ageri da)

```
Sartu zenbaki bat: 133
Blum-en zenbakia da. Beste zenbakiren bat aztertu nahi
al duzu? b
Sartu zenbaki bat: 7
Ez da Blum-en zenbakia. Beste zenbakiren bat aztertu
nahi al duzu? e
Sakatu tekla bat amaitzeko.
```

Pantaila